Corrección de Iluminación Espacial mediante Ajuste Polinomial con Máscara

```python
import numpy as np
import spectral as spy
from scipy.linalg import lstsq

# ----------------------------
# 1) Cargar cubo
# ----------------------------
img = spy.open_image("imagen.hdr")
cube = img.load().astype(np.float64)

rows, cols, bands = cube.shape

# ----------------------------
# 2) Crear coordenadas NORMALIZADAS
# ----------------------------
y, x = np.mgrid[0:rows, 0:cols]
x = x / cols
y = y / rows

# ----------------------------
# 3) Crear m scara
# ----------------------------
# Usamos una banda intermedia como referencia
band_ref = cube[:, :, bands // 2]

# Seleccionamos p xeles por encima del percentil 60
threshold = np.percentile(band_ref, 60)
mask = band_ref > threshold

# ----------------------------
# 4) Funci n de correcci n con m scara
# ----------------------------
def correct_band(band):

    # Aplicar m scara
    x_masked = x[mask]
```

```
    y_masked = y[mask]
    z_masked = band[mask]

    # Construir matriz de dise o SOLO con p xeles
       v lidos
    X_masked = np.column_stack([
        np.ones(x_masked.size),
        x_masked,
        y_masked,
        x_masked**2,
        y_masked**2,
        x_masked*y_masked
    ])

    # Ajuste por m nimos cuadrados
    coeffs, _, _, _ = lstsq(X_masked, z_masked)

    # Construir modelo completo
    X_full = np.column_stack([
        np.ones(rows*cols),
        x.ravel(),
        y.ravel(),
        x.ravel()**2,
        y.ravel()**2,
        (x*y).ravel()
    ])

    illumination = (X_full @ coeffs).reshape(rows, cols)

    # Evitar divisi n por cero
    illumination[illumination == 0] = 1e-8

    corrected = band / illumination

    return corrected

# ----------------------------
# 5) Aplicar banda por banda
# ----------------------------
corrected_cube = np.zeros_like(cube)
```

```
76
77 for b in range(bands):
78     corrected_cube[:, :, b] = correct_band(cube[:, :, b
       ])
```